DEO.WORKS

MERCHANT STABLECOIN CHECKOUT & SETTLEMENT INFRASTRUCTURE

Public review pack

An expanded, evidence-led account of the build and its next milestone

BASE SEPOLIA FIRST / TESTNET ONLY / SEPTEMBER 2026

DEO.WORKS / PUBLIC REVIEW EDITION

READING MAP

An honest view of an ambitious build

This pack separates built and tested foundations, publicly inspectable Base Sepolia proof, the complete Base-only demo target, and longer-horizon opportunities. It is designed for merchants, integration partners, developers and independent reviewers.

01 A public review of what DEO WORKS is building

02 The commercial thesis

03 Three layers of evidence

04 What the customer should experience

05 What the merchant should experience

06 The payment and fee model

07 Why the settlement router matters

08 Recorded Base Sepolia proof

09 What was exercised before the Base focus

10 The engineering underneath the checkout

11 Consent, timing and one-use behavior

12 Independent chain evidence and reconciliation

13 The multi-merchant, multi-store demo target

14 The integration contract for a store

15 How a Base checkout should resolve

16 What the first public demo must prove

17 A staged route from foundations to demo

18 Base first, then a measured multichain path

19 Product opportunities beyond payment acceptance

20 A reviewer-friendly evidence checklist

21 Boundaries worth keeping visible

22 Technical appendix: example payment accounting

23 Technical appendix: state and evidence

24 Sources, public identifiers and contact

THESIS AND EVIDENCE / 01 OF 24

A public review of what DEO WORKS is building

DEO WORKS is building merchant-facing stablecoin checkout and settlement infrastructure. The immediate product objective is a complete, publicly reviewable Base Sepolia testnet purchase: a merchant registers, connects one or more stores, a customer pays with a wallet, and one successful payment sends the merchant's net amount to the merchant's settlement wallet and the disclosed platform fee to the DEO WORKS treasury. The checkout, receipts and merchant-facing status must agree with the chain. No one should need to interpret a transaction hash as proof of completed settlement.

That is the destination of the current build, not a claim that a public end-to-end Base checkout is already live. Important foundations exist: merchant onboarding and wallet-control components, a testnet checkout integration on an earlier network, a deployed Base Sepolia settlement-router proof, extensive deterministic fee and contract tests, and a carefully staged authorization/evidence architecture. The remaining work is to join those foundations into a trustworthy, durable, multi-merchant product and then measure the experience with real testnet users.

The near-term sequence is deliberately practical: Base-only functional demo; actual measurement of checkout behavior; one independently verified origin-to-Base route if it can meet the product's UX and evidence standards; then wider expansion. Base is the first settlement destination and proving ground. This order leaves room for a larger multichain vision without disguising a bridge delay or an incomplete route as a finished retail checkout.

> **PUBLIC STATUS | This document describes a project under construction. Base Sepolia USDC and ETH are testnet assets with no financial value. The public Base checkout, merchant-fee collection and multichain paths described as targets are not represented as live services.**

THESIS AND EVIDENCE / 02 OF 24

The commercial thesis

Many webstores sell in more than one channel, manage multiple storefronts, and want payment operations that do not force them to replace their commerce stack. A stablecoin option becomes commercially meaningful only if it behaves like a merchant product: predictable order binding, understandable fees, an intelligible customer handoff, settlement to the right merchant, and a reliable answer to the question "was this order actually paid?"

DEO WORKS is designed as infrastructure that can sit beside a store's existing checkout rather than as a new marketplace that must own the entire shopping journey. The store remains the source of truth for its products, cart and order total. DEO WORKS focuses on merchant identity, payment intent, wallet consent, settlement rules, transaction observation and evidence that can be reconciled against the order. That division is important: a visually polished payment button is not enough if the merchant cannot identify the purchaser, match the exact amount, or understand a delayed transaction.

The intended value to a merchant is not merely "accept crypto." It is an additional, auditable testnet payment rail built around a known stablecoin and a clear merchant-net/platform-fee split, with room for several stores under the same merchant account. The intended value to a customer is a recognizably ordinary buying moment: review the amount, see what wallet and network are required, consent, submit from the customer's wallet, and receive a truthful status. The intended value to partners is an integration that makes its boundaries and evidence inspectable rather than asking them to trust a slide deck.

Commercial success is still to be measured. We do not claim production conversion rates, savings, volume, revenue, regulatory coverage or merchant adoption. The testnet demo is intended to produce the first credible observations of usability and operational behavior.

THESIS AND EVIDENCE / 03 OF 24

Three layers of evidence

The project is easiest to evaluate when three distinct claims are kept separate.

First, there is implemented engineering: merchant registration and authentication components; merchant wallet challenge and replacement flows; a hosted earlier-network testnet storefront with a real wallet-to-merchant token-transfer path; a Solidity settlement router and contract tests; deterministic fee calculation; and a substantial backend body of authorization, consent, timing, replay, historical-policy and failure-handling work. These are meaningful assets. They are not a substitute for the missing public Base integration.

Second, there is a public Base Sepolia chain proof: a settlement-router deployment recorded at the address in this pack. A deployment verifies that contract code was put on the intended test network; it does not by itself prove that merchants can sign up, install the widget, take a payment, split fees, receive a reconciled receipt or recover from ambiguous outcomes. The router was designed to start paused. Its activation for a merchant rail remains separately gated.

Third, there is the complete product target: multiple merchants, multiple stores per merchant, real testnet customers using their own wallets, exact merchant-net and treasury-fee settlement in one successful on-chain operation, and durable statuses backed by independent chain evidence. That is what we mean by a functional public demo. Anything short of that should be labeled a component demonstration or an internal diagnostic, not advertised as a finished payment service.

> **REVIEW RULE | An approved design is not implemented code. A passing internal test is not a live payment. A submitted transaction is not a confirmed purchase. A deployed contract is not a complete merchant integration.**

PRODUCT EXPERIENCE / 04 OF 24

What the customer should experience

The intended checkout begins at a real store order, not at an arbitrary token-transfer screen. The merchant's store supplies a server-authoritative order total, a store identity, and a stable order reference. The payment service creates a bounded checkout whose amount, token, selected network, recipient, platform fee, payer terms and expiry are frozen. The customer sees the gross cart amount and the payment steps before consenting.

The wallet stays under the customer's control. In the first Base-only demo, the payer uses an eligible wallet on Base Sepolia with test USDC and enough test ETH for gas. A network mismatch or insufficient balance should be explained before a doomed submission wherever possible. The wallet presents the action; the customer can approve or reject it. DEO WORKS does not receive the wallet's private key and cannot manufacture the customer's consent.

The checkout then moves through states that a human can understand: prepared, awaiting wallet action, submitted, observed on-chain, confirmed under the selected testnet policy, or failed/unresolved. A rejected signature is not a payment. A mere transaction hash is evidence of submission, not evidence of successful execution. A failed receipt or the wrong token, recipient or amount cannot become a paid order. If observation is temporarily incomplete, the product should say so and retain the pending state rather than rewarding optimism with a false success screen.

This is the usability standard we want to test in public. The demo must expose clear next steps after wallet rejection, chain delay, refresh and retry. It must work on the storefronts selected for validation without relying on a presenter manually editing database records or narrating away errors. Mobile wallet handoff, accessibility and wording deserve real user testing; they are product acceptance criteria, not decorative polish.

PRODUCT EXPERIENCE / 05 OF 24

What the merchant should experience

One person may operate several webstores. The intended account model permits many merchants and lets each merchant create and manage multiple stores, without a fixed product-level limit imposed by this demo design. Each store belongs to one merchant. That ownership must be enforced by the backend, not inferred from a front-end URL or editable form field. The merchant manages their stores and checks order/payment status from an authenticated context.

For the initial Base demo, the settlement destination is one active, verified wallet per merchant per enabled network. This is a deliberate simplification, not a permanent ban on store-specific wallets. It lets a merchant operate several stores while preserving a clear recipient rule for every checkout. When the product expands to another EVM network or a non-EVM network, wallet enrollment can be handled per network and technology family rather than assuming every address works everywhere. Optional store-level settlement overrides remain a later, separately designed capability.

Wallet changes require more than changing a text field. A newly verified wallet must agree with the currently eligible payment-rail enrollment before new checkouts are issued to it. If a merchant replaces or revokes wallet A in favor of wallet B while the rail still binds A, new issuance must pause until a successor enrollment reconciles the two. Existing orders retain the recipient and expiry that were frozen when they were created; no historical payment should be silently rewritten or replayed into B.

The intended merchant dashboard shows the order, gross amount, merchant net, platform fee, chain/network, frozen recipient, transaction links and honest status. It must distinguish a payment awaiting evidence from a confirmed one. A merchant should not need to reverse-engineer a block explorer to know which store and order a receipt belongs to.

PRODUCT EXPERIENCE / 06 OF 24

The payment and fee model

The approved testnet fee policy is merchant-borne. The customer pays the store's gross cart amount; DEO WORKS does not add a separate platform surcharge to the customer's price. On a successful eligible router payment, the gross amount is divided into the merchant's net amount and the disclosed DEO WORKS fee. Both legs belong to the same atomic on-chain execution: either the configured transfer succeeds as a whole or it does not produce a partial successful settlement.

The normal testnet-policy platform rate is 40 basis points, equal to 0.40% of gross. The approved policy ceiling is 200 basis points, equal to 2.00%. A bounded promotion can reduce the normal fee, including a reduction as high as 99.99%, subject to its reference and validity window. An ineligible fee calculation that rounds to zero cannot be silently rounded up to collect extra value. The fee is calculated once with integer test-USDC units and a single final floor, avoiding floating-point drift.

For example, a 10.00 test-USDC order at the normal 0.40% rate yields 0.04 test USDC in platform fee and 9.96 test USDC merchant net. This is an illustration of the approved arithmetic, not evidence of a production transaction. Actual testnet execution also requires the customer's gas budget in test ETH; that gas is separate from the merchant-borne platform fee and should be explained in the checkout. Merchant commercials, taxes and future live-chain terms require their own approval and disclosure before any mainnet service.

The contract checks a signed split of gross, merchant net and fee, but the higher-level service is responsible for applying the rate ceiling and promotion policy before authorizing it. That is why rate evaluation, exact checkout snapshot and the on-chain transfer must be considered together. A contract that can move tokens is not a pricing policy engine by itself.

PRODUCT EXPERIENCE / 07 OF 24

Why the settlement router matters

The local SettlementRouterV1 contract is designed for an exact, single-network stablecoin purchase. The customer remains the payer; the merchant and DEO WORKS treasury are the two recipients. It does not hold a rolling customer balance, perform a swap, route across chains, or act as a general bridge. For an eligible checkout it attempts the defined token transfers as one atomic transaction and emits evidence the backend can compare with the frozen order.

This is a strong fit for the first Base demo. It removes a second, discretionary payout operation between "customer paid" and "merchant received net" while making the fee leg visible in the same execution. It also gives the observer a specific event and transfer pattern to verify. The design is more precise than a generic request to transfer stablecoins to an address, yet intentionally smaller than a universal payment protocol.

The router's existence and deployment do not authorize its use. Before any public checkout, the relevant merchant enrollment, token, router, recipient, treasury and fee facts must be current and mutually consistent. The signature/coordinator and actual transaction-submission path must be integrated and independently tested. A signed intent should bind one exact purchase, not become a reusable merchant allowance or a license to change recipients later.

The code and tests have been developed around these boundaries, including the difference between a customer wallet submitting a transaction and the backend merely preparing or observing one. The public demo will prove the complete chain only when a customer can go from store cart to independently reconciled merchant-net and treasury-fee receipt.

BASE PROOF AND FOUNDATIONS / 08 OF 24

Recorded Base Sepolia proof

The documented Base environment is Base Sepolia, chain ID 84532, with test ETH used for gas. Circle's published Base Sepolia test-USDC contract is 0x036CbD53842c5426634e7929541eC2318f3dCF7e. These network and asset facts should be pinned and rechecked when the public checkout is activated; a token symbol alone is not sufficient evidence of the right asset.

The project's recorded testnet settlement-router deployment is 0x119f9343c046663b0d47eb58e3da51782a99d878. The recorded deployment transaction is 0x69c441ef419dbc4e99eba74360481213eef2c74bea3489f83ac7c59c2533a2b3, and the recorded runtime bytecode hash is 0xd08b728336f1e6301dc64b3899451d1bfab955e713c7af0b4c61e4290558cb1f. The deployment record notes an initially paused contract. Reviewers can inspect the explorer links listed in the evidence appendix.

The significance of the proof is specific. It demonstrates a Base Sepolia contract deployment and gives public identifiers against which the on-chain code can be checked. It does not establish live merchant enrollment, a functioning Base widget, actual router payments, fee receipts, or a production operating environment. Those remain the work of the end-to-end demo checkpoint.

There is no claim here of Base endorsement, funding, official integration status or approval. Base is the chosen first settlement network for this testnet build; DEO WORKS is an independent project building on public infrastructure.

BASE PROOF AND FOUNDATIONS / 09 OF 24

What was exercised before the Base focus

An earlier testnet storefront integration exercised a direct customer-wallet-to-merchant token-transfer flow on Arc Testnet. That work included merchant-facing onboarding elements, a hosted demo storefront, a wallet checkout and backend verification of the direct transfer. It was a useful proof that the team could connect a store, customer wallet and observable testnet payment. It was not a router-based split settlement and was not treated as a public production release.

The Base-focused plan reuses the useful lessons and infrastructure while raising the standard. The earlier transfer was on one network, straight to a merchant; the intended Base demo requires a real fee split, merchant/store tenancy, durable order linkage, recipient agreement, exact consent and honest post-submission reconciliation. A result on one testnet cannot simply be relabeled as a finished result on another. Network configuration, asset contracts, gas, receipts and wallet UX all need direct verification on Base Sepolia.

This distinction matters commercially. A small private demonstration is legitimate progress, but a public merchant product needs to survive visitors who are not guided by the team. They will reject a wallet prompt, close a tab, retry, arrive with the wrong network and ask whether a delayed order is paid. The Base demo is defined to answer those cases in product behavior, not in an operator's explanation after the fact.

BASE PROOF AND FOUNDATIONS / 10 OF 24

The engineering underneath the checkout

The project has invested heavily in the less visible work that makes a payment believable. Merchant identity and wallet-control primitives, canonical input handling, integer fee calculations, a contract interface, replay and timing rules, explicit consent, historical policy association, guarded policy storage and uncertain-outcome handling are separated into checkpoints. This structure is intentional: it avoids letting a single front-end success event decide that funds moved correctly.

The current internal authorization and evidence components include extensive synthetic and database-backed tests. The latest implemented policy repository records immutable reviewed policy history and handles guarded creation, historical acknowledgement, concurrency, expiration and reporting uncertainty. Its output is permanently non-authorizing. It does not sign, submit a transaction, move funds or turn a checkout live. Subsequent execution-signing and submission/observation contracts are being closed in separately scoped implementation steps rather than being smuggled into a diagnostic result.

That investment is commercially relevant even though it is not itself a customer-facing screen. A merchant cannot reconcile an order if the system loses the exact original recipient, confuses a historical record with a current authorization, counts an approval as payment, or treats a database timeout as permission to retry blindly. The careful internal work addresses those classes of failure in advance of the public integration.

The remaining challenge is equally concrete: provide trusted live evidence and operational controls, connect the approved invariants to a bounded signing/transaction path, integrate the store and merchant APIs, and persist final status after actual chain observation. Internal checkpoint depth should be celebrated as progress without pretending that it closes those operational gaps.

TRUST AND INTEGRATION / 11 OF 24

Consent, timing and one-use behavior

A customer should authorize one intelligible purchase, not an abstract future power to spend. The intended checkout binds the merchant, store, order, payer, Base network, canonical test-USDC token, router, gross amount, fee, merchant net, recipient and expiry. The wallet action and the service's authorization facts must refer to the same frozen purchase. A later merchant wallet change must not alter a prior consent.

Timing is not a cosmetic field. The system's reviewed rules distinguish the original checkout expiry from the freshness of the evidence that supported it. Freshness can lapse while the team awaits external operations; it must be checked again at the final decision point. Historical acknowledgement of a valid prior record is not a new permission to execute it after expiry. A denied, revoked or uncertain condition needs to remain visible as such rather than being converted into a success by another request.

One-use protection is also about side effects, not only about a unique ID. If a first submission may already have reached the network, a timeout cannot justify constructing a second payment from the same purchase. The design retains the original attempt and requires observation or resolution. An ambiguous outcome is a reason to stop and investigate, not to debit the customer again for a cleaner-looking receipt.

The demo's acceptance tests must therefore include wrong merchant and store, wrong payer, wrong network or token, stale information, expired or replaced wallet, duplicate checkout, repeated signature, delayed receipt, failed receipt and inconsistent chain evidence. Positive cases are necessary too: a legitimate customer purchase should complete with the exact amount and recipients under the selected testnet confirmation rule.

TRUST AND INTEGRATION / 12 OF 24

Independent chain evidence and reconciliation

The back end must observe what actually happened on Base Sepolia, not accept a browser's declaration that the wallet sent a payment. A submitted transaction hash is a locator. A successful receipt, expected contract, expected token, expected event and expected transfer legs are evidence. Those facts must be compared with the immutable checkout: merchant, store, order, payer, token, gross, merchant net, platform fee and recipient.

The selected demo confirmation policy requires a successful matching router receipt followed by additional Base blocks and repeated checks of the receipt and block identity before the system calls the purchase confirmed. That is a bounded testnet operational rule, not a claim of irreversible Ethereum finality. If the chain reorganizes, the receipt disappears, or any expected field fails to match, status cannot remain blindly "paid." The merchant and customer should see a truthful pending, failed or unresolved state.

Order fulfillment must be linked to this verified state. The store cannot treat wallet approval, token approval, popup closure or a hash pasted by the customer as a completed sale. Conversely, a delayed but valid receipt should be recoverable after refresh: evidence persists and the right merchant/store order can be reconciled without asking the customer to pay again.

This is where payment infrastructure earns its name. It is not enough for assets to move eventually; the service must connect a specific purchase to a specific on-chain outcome, retain that relationship, and explain it to both sides of the transaction.

TRUST AND INTEGRATION / 13 OF 24

The multi-merchant, multi-store demo target

The demo is designed for many merchant accounts, each able to attach multiple stores. For a manageable validation fixture, the team can exercise at least two distinct merchants and three stores - for example, two stores owned by merchant A and one owned by merchant B. Those numbers are test cases, not product caps or a claim that three outside merchants are enrolled today.

Every store must have stable ownership and its own identity. Credentials and API requests from one merchant must not see or alter another merchant's orders. A store should not be able to substitute another merchant's settlement wallet, fee policy or order reference. Multiple stores under one merchant can share that merchant's verified Base wallet in the first demo, while still producing separate store-level order history and metrics.

The widget or integration layer should be installable across the selected demo storefronts without rebuilding the payment rules for each storefront. The merchant's commerce system remains authoritative for product catalog and checkout total; DEO WORKS receives a bounded, authenticated purchase request and returns only the safe public fields needed by the customer. The same backend rules must apply regardless of storefront theme, page layout or device.

The success condition is end-to-end, not a collection of screenshots: register merchant; verify and enroll wallet; create stores; install integration; place real demo orders from customer wallets; execute Base test-USDC settlement; observe exact merchant-net and treasury-fee legs; surface durable statuses to the correct merchant and customer. Negative tests - rejected signature, cross-tenant access, changed order amount, wrong recipient and ambiguous receipt - must fail closed.

TRUST AND INTEGRATION / 14 OF 24

The integration contract for a store

An integration-friendly checkout starts with a narrow responsibility split. The storefront knows its catalog, customer session, taxes and fulfillment rules. The payment service knows the merchant/store identity, approved rail, frozen order total, recipient, token, fee snapshot and observation record. A server-to-server call should establish the checkout before exposing wallet instructions to the browser. Browser-controlled prices or merchant IDs should never determine the final payable amount.

The public widget should be small enough to install on several storefronts but precise enough to show the right chain, token, total, wallet status, consent action and progress. It should not silently switch networks or hide a gas requirement. An integration can provide a callback or polling view for order status, but only the backend's independently reconciled state should authorize fulfillment. The merchant also needs a way to inspect an individual payment, not just an aggregate count.

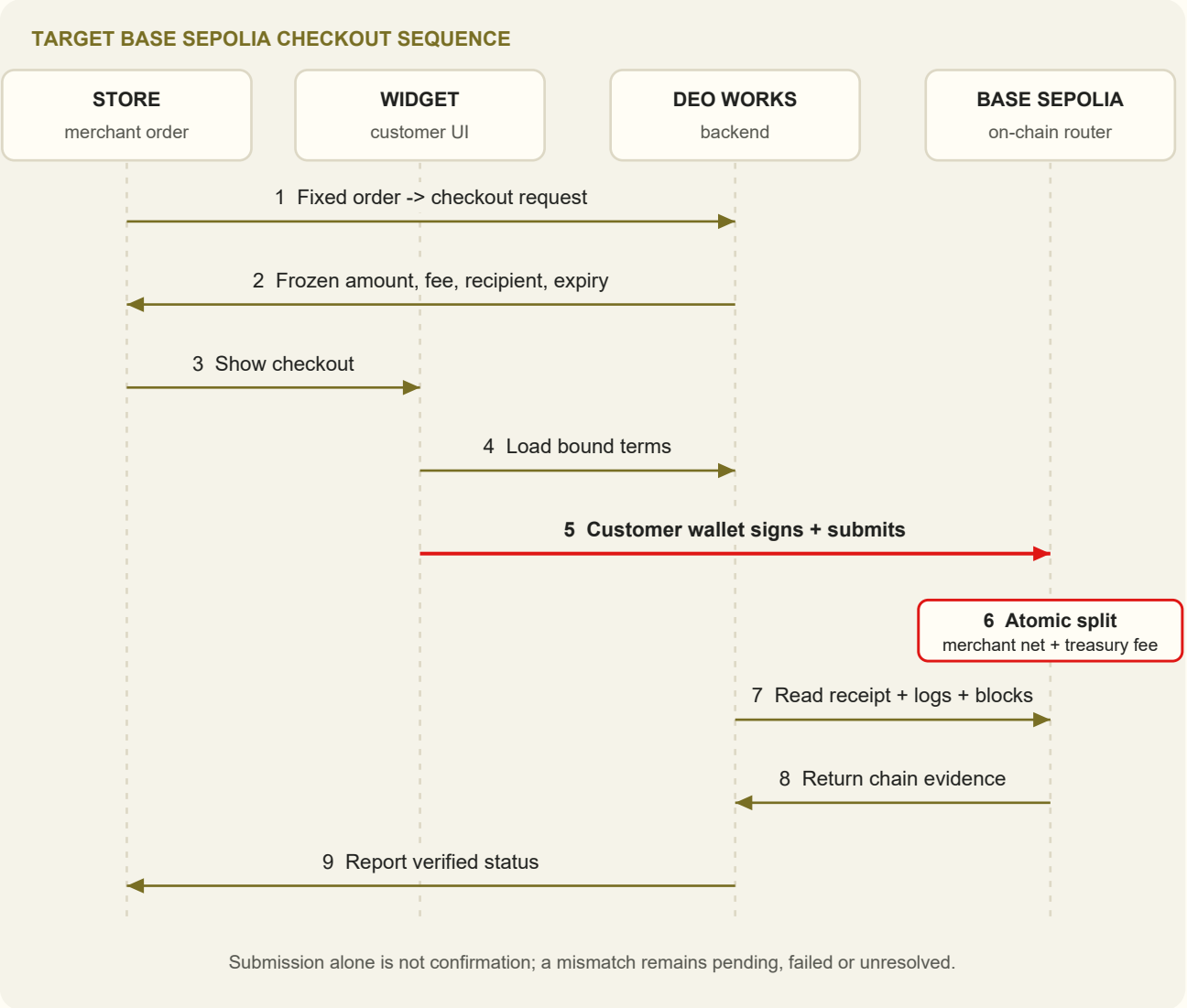
Merchant keys and customer wallet controls have different roles. An API credential authenticates a merchant/store integration; it is not authority to spend a customer's wallet. A customer signature approves one bounded transaction; it is not an API key or a general merchant mandate. The first demo should make those scopes understandable and keep server-only credentials out of the browser.

We expect different stores to have different design systems. The integration therefore should preserve store branding and cart context while keeping the payment facts legible. The current demo-store foundation is useful for proving this approach, but the complete multi-store Base payment loop still has to be integrated and tested.

TRUST AND INTEGRATION / 15 OF 24

How a Base checkout should resolve

The sequence below describes the target public demo, not a flow already available to outside customers. The store owns the order total; the DEO WORKS backend freezes the checkout and later verifies chain evidence; the customer signs and submits through their own wallet; SettlementRouterV1 executes on Base Sepolia. In the diagram, "Base Sepolia / Router" is the on-chain component, not a backend service. A submitted transaction remains pending until the backend independently confirms the exact receipt and settlement legs.



THE FUNCTIONAL DEMO / 16 OF 24

What the first public demo must prove

We will call the Base Sepolia release a functional public demo only when an unaffiliated visitor can perform a genuine testnet purchase from at least one of the chosen stores without a developer standing beside them. The merchant must be able to sign in, inspect the corresponding order and see the correct net settlement; the DEO WORKS fee must be visible at the designated treasury as part of the same successful router operation. A second merchant and multiple stores must show that tenant ownership is real rather than hard-coded to one fixture.

The demo should include a clear setup path for a test wallet, test USDC and test ETH; honest network switching and wallet rejection handling; usable mobile behavior; and a status page that can survive reload. The team should be able to repeat the flow with distinct accounts and orders, demonstrate failures safely, and point a reviewer from checkout ID to chain receipt to merchant order.

Release evidence should include a reproducible test matrix, independent receipt verification, known limitations, and operating controls for pause or disable. The quality bar is not zero latency or perfect completion. It is truthful state, no false settlement, no double charge from retry, no cross-merchant leakage, and no hidden manual step in the advertised purchase path.

This target is demanding because it is a real product demonstration rather than a contract call in a terminal. It is also narrow: Base Sepolia only, canonical test USDC only, selected demo stores, no swaps and no cross-chain dependencies. Those choices give the project a credible first public proof that can be measured.

THE FUNCTIONAL DEMO / 17 OF 24

A staged route from foundations to demo

The first workstream closes the remaining internal execution-signing and observation contracts without making diagnostics accidentally executable. Synchronous, supplied-snapshot checks can be useful internal slices, but trusted live evidence, one-time signing coordination and actual transaction observation require their own explicit owners and tests. No internal positive diagnostic is a production signing key.

The second workstream adds the operational Base merchant/store purchase service. It must connect existing merchant registration and wallet proof to network-specific rail enrollment; authenticate tenant and store; freeze a server-authoritative order; compute the fee once; bind the recipient; and persist each attempt and status. Wallet replacement has to reconcile with enrollment before new issuance, while older checkouts keep their original terms.

The third workstream connects customer wallet submission to the Base router and tests exact test-USDC settlement. The customer, not a hidden backend custodian, controls the paying wallet. The service may prepare bounded data and later observe the chain, but may not claim success from a preparatory approval, signature or submitted hash.

The fourth workstream completes independent receipt verification, merchant/customer reporting, a pause/incident path, multiple storefront integrations and usability measurement. A staged private pilot may precede the public demo, but the public claim must describe exactly what users can do at that time. Each workstream has an acceptance gate; a partially finished earlier gate is not washed away by an attractive front end.

THE FUNCTIONAL DEMO / 18 OF 24

Base first, then a measured multichain path

DEO WORKS ultimately wants customers to be able to originate payments from more than one network while merchants can choose a settlement destination, with Base as an important settlement layer. That direction is commercially attractive, but cross-chain checkout is not just a network selector. It introduces source-chain finality, bridge or messaging trust, destination execution, liquidity and asset mapping, longer and variable waiting time, failed or delayed attestation, and a second chain's evidence. All of those affect what the customer and merchant should be told.

For that reason the first public demo is Base-only. We will measure the actual wallet-to-receipt journey and identify the UX budget for a retail purchase. After that, one candidate origin-to-Base route can be evaluated against official deployments and then tested live. The route must demonstrate a specific source transaction, authenticated cross-chain message, destination receipt and exact merchant settlement, with pending and failure states that remain comprehensible. A documentation support matrix alone does not prove a usable checkout time.

Arc and Unichain are plausible future testnet origin candidates; neither is being advertised here as an operating DEO WORKS-to-Base payment lane. Robinhood and any other candidate require the same route-by-route proof rather than a logo on a roadmap. A non-EVM route, such as a future Solana path, would require a dedicated adapter and wallet/asset model instead of reusing EVM assumptions. This is an expansion plan, not a promise of immediate chain coverage.

THE HORIZON AND ITS LIMITS / 19 OF 24

Product opportunities beyond payment acceptance

A dependable checkout rail could support several product surfaces over time. A merchant could use a lightweight widget in an existing webstore, a hosted payment link, or a point-of-sale screen that presents a bounded amount and QR code for a customer wallet. The essential requirement in each case is the same: the amount and recipient must be bound to a real merchant and the product must wait for verified settlement before saying "paid." A QR image alone is not a receipt.

There is also a discovery opportunity. If several independent merchants can onboard stores and publish eligible products, a customer-facing discovery layer could direct buyers to those stores while preserving merchant ownership of fulfillment. This would be a separate product and growth decision; no customer marketplace or mobile app is represented as launched in this pack. The near-term infrastructure should, however, avoid technical choices that make future discovery or point-of-sale impossible.

Commercial packaging can follow measured usage. Potential partner value includes integration support, verifiable fee accounting, checkout observability and merchant-level operation across several stores. Pricing, service levels, geographic eligibility and legal obligations must be decided with actual operating evidence. DEO WORKS is inviting technical review and expressions of interest, not offering a production payment processing agreement through this PDF.

THE HORIZON AND ITS LIMITS / 20 OF 24

A reviewer-friendly evidence checklist

The most productive review questions are concrete. Can the team show the deployed Base Sepolia router, identify the exact test-USDC contract, and distinguish contract deployment from an activated merchant rail? Can it explain the fee split in integer units and show where the merchant receives net and the treasury receives fee? Can it show how a checkout is bound to a merchant, store, order, payer, recipient, token, network and expiry?

Can the team demonstrate that a browser cannot mark its own order paid; that a transaction hash, approval or missing receipt does not settle an order; that wrong recipient or amount is rejected; and that a delayed observation survives a refresh? Can two merchants operate separate stores without sharing orders or credentials? If a wallet is replaced, do new checkouts pause until enrollment matches the new verified recipient while historical checkouts retain the old one?

The review should also ask what happened in real tests, not just what the architecture intends. How many distinct test wallets and merchants completed the flow? What were the observed wallet-rejection, gas, confirmation and mobile-handoff problems? How long did a real checkout take from consent to truthful confirmation? Which incidents required operator attention? The public testnet demo exists to answer those questions with evidence before stronger commercial claims are made.

The team welcomes architectural feedback, partner integration questions and merchant expressions of interest. A public reviewer should be able to trace a claim to a test, document or chain record, and to recognize when a line is explicitly still a target.

THE HORIZON AND ITS LIMITS / 21 OF 24

Boundaries worth keeping visible

The project is not presenting a stablecoin checkout as a bank account, a paused contract as an active payment rail, or internal policy diagnostics as live authorization. It does not claim a security audit by an independent auditor unless one is later published. It does not claim production uptime, conversion, revenue, active merchant count, a Base partnership, grant award or permission to operate in every jurisdiction.

The non-custodial design goal is precise rather than magical. The customer submits from their own wallet, and the planned router sends the defined token amounts directly to recipients in one execution. This reduces certain custody and payout risks; it does not remove smart-contract risk, wallet compromise, RPC failure, token risk, consumer protection duties, tax issues or every form of operational responsibility. Testnet correctness is a prerequisite to, not proof of, mainnet suitability.

No cross-chain service is part of the initial Base-only checkout. No pricing claim compares DEO WORKS with card processors on a fully loaded basis. No automatic refunds, chargebacks, exchange-rate conversion or fiat off-ramp is claimed. Refunds and exceptional merchant disputes need separate policy and implementation before any production offer.

These boundaries are not a retreat from ambition. They protect the difference between what has been built, what can be shown next, and what would require a broader set of product, security and regulatory decisions. That clarity is part of a credible commercial story.

TECHNICAL APPENDIX / 22 OF 24

Technical appendix: example payment accounting

The on-chain asset for the initial demo is Circle's canonical Base Sepolia test USDC with six decimal places. Suppose a store creates an order for 10.000000 test USDC. In integer units, gross is 10,000,000. At 40 basis points with no promotion, the platform fee is floor(10,000,000 x 40 / 10,000) = 40,000 units, or 0.040000 test USDC. Merchant net is 9,960,000 units, or 9.960000 test USDC. The successful router payment must account for exactly 10,000,000 units in total across the merchant and treasury legs.

With a promotion, the approved arithmetic first applies its bounded discount to the normal fee rate and floors only once at the end in integer units. The fee cannot exceed the ceiling, and an ineligible zero fee is not repaired by taking one unit from the merchant. The backend's fee snapshot should retain the selected normal rate, discount, effective calculation, gross, fee, net, policy reference and expiry so the signed and observed purchase can be reconciled later.

This example illustrates the accounting in token units and the evidence a reviewer should be able to inspect; it is not a live commercial invoice. If the checkout eventually moves to mainnet, real-asset, gas, tax, disclosure and settlement questions must be revisited under separate approval.

TECHNICAL APPENDIX / 23 OF 24

Technical appendix: state and evidence

A robust purchase has at least four distinct records: the merchant/store order, the frozen payment intent, the wallet/transaction attempt, and the independent chain observation. These records should share stable identifiers and never rely on a mutable browser display as their common source of truth. The order amount is fixed before the wallet prompt; the recipient and fee are selected from current eligible merchant enrollment and approved pricing; the attempt records what was actually submitted; the observer records what the chain actually executed.

The evidence needed to mark a purchase confirmed includes the correct chain, canonical token, configured router, successful receipt, exact checkout association, expected payer and recipient, and exact merchant-net and treasury-fee amounts. The selected Base testnet confirmation policy adds successor blocks and repeated receipt/block-identity checks. A wallet approval transaction cannot meet this condition. A transaction that reverts or sends the right amount to the wrong merchant cannot meet it either.

The uncertain branch is first-class. If an external call times out, a receipt is temporarily unavailable, or reporting is interrupted, the checkout retains its prior attempt and is observable as pending or unresolved. It is not reissued as a fresh authorization simply to simplify the UI. That rule connects the deep internal checkpoint work to an ordinary promise the merchant can understand: no invented success and no silent second debit.

TECHNICAL APPENDIX / 24 OF 24

Sources, public identifiers and contact

Base's documentation lists Base Sepolia as chain ID 84532: docs.base.org/get-started/connect-to-base . Circle's official USDC contract-address page lists the Base Sepolia test token: developers.circle.com/stablecoins/usdc-contract-addresses . Network and token details were checked for this edition in September 2026 and should be checked again before activation.

Project-recorded Base Sepolia router address: 0x119f9343c046663b0d47eb58e3da51782a99d878 . Project-recorded deployment transaction: 0x69c441ef419dbc4e99eba74360481213eef2c74bea3489f83ac7c59c2533a2b3 . Explorer entry points: sepolia.basescan.org/address/0x119f9343c046663b0d47eb58e... and sepolia.basescan.org/tx/0x69c441ef419dbc4e99eba743604812... . The address and transaction are deployment evidence, not proof of public checkout traffic.

The current public website is deo.works . Its documents and video are accessible from the home page. For technical review and project correspondence: info@deo.works . For a non-binding merchant or integration-partner expression of interest: partners@deo.works and deo.works/partner/ . Please do not send wallet recovery phrases, private keys, payment credentials or sensitive customer information.

DEO WORKS is building the checkout and settlement layer one verifiable step at a time. The nearest milestone is not a promise about every chain or every retailer. It is a complete, honestly measured Base Sepolia testnet purchase that a real merchant and customer can understand and that an independent reviewer can inspect.